

THE CLAIMS

1. (Currently Amended) A system for allocating a resource in a network comprising: first logic in the network for determining if any of a plurality of persistence policies, comprising at least one first persistence policy and at least one second persistence policy, is applicable to a service request, and, if so, allocating the resource to the request based on application of the persistence policy determined to be applicable;~~and~~

the first logic configured so that:

when a first type of service request is received, the first logic determines if the at least one first persistence policy is applicable;

when a second type of service request is received, the first logic determines if the at least one second persistence policy is applicable;

when the first type of service request is received but it is determined that the first persistence policy is inapplicable, the first logic determines if the at least one second persistence policy is applicable; and

second logic in the network for allocating the resource to the request based on application of a load balancing policy only if none of the plurality of persistence policies is determined to be inapplicable as determined by the first logic;

~~wherein the persistence policy, if applicable, overrides the load balancing policy by controlling the allocation.~~

2. (Original) The system of claim 1 wherein the first logic determines if a persistence policy is applicable to a service request having an originator through consideration of whether or not an allocation exists or recently expired for the originator the service request.

3. (Currently Amended) A system for allocating a resource, in a network, to a resource request, the request having an originator, based on application of a persistence policy comprising:

first logic in the network for determining whether an allocation exists or recently expired for the originator of the resource request, based on application of any of a plurality of persistence policies, comprising at least one first persistence policy and at least one second persistence

policy, to the request, and, if so, identifying the resource which is the subject of the existing or recently expired allocation; and

the first logic configured so that:

when a first type of service request is received, the first logic determines whether an allocation exists or recently expired for the originator of the resource request by applying the at least one first persistence policy, or the at least one second persistence policy when the at least one first persistence policy is determined to be inapplicable;

when a second type of resource request is received, the first logic determines whether an allocation exists or recently expired for the originator of the resource request by applying the at least one second persistence policy; and

second logic in the network for allocating the resource, once identified, to the resource request.

4. (Original) The system of claim 3 wherein the resource request is derived from or represented by a packet.

5. (Withdrawn) A system for maintaining a data structure useful for allocating a resource to a resource request based on application of a persistence policy comprising:

first logic for making an entry in the data structure representing an allocation, upon or after implementation of the allocation, and time-stamping the entry with a time-stamp indicating the time when or about when the allocation is terminated; and

second logic for scanning the data structure, and deleting entries for which a time-out condition is determined to exist.

6. (Withdrawn) The system of claim 5 wherein the second logic has access to a current time, and determines that a time-out condition exists if the time-stamp value equals or exceeds the current time by a predetermined amount.

7. (Withdrawn) The system of claim 5 wherein the second logic repetitively scans the data structure.

8. (Withdrawn) The system of claim 5 wherein the second logic periodically scans the data structure.

9. (Withdrawn) The system of claim 6 wherein the predetermined amount is programmable.

10. (Withdrawn) A system for making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the system comprising:

first logic for deriving a first index from information relating to the resource request if such information is available, and using the first index to make an entry in the data structure representing the allocation if the first index is available; and

second logic for deriving a second index from information relating to the resource request, and using the second index to make an entry in the data structure representing the allocation.

11. (Withdrawn) The system of claim 10 wherein the data structure is a history table.

12. (Withdrawn) The system of claim 10 wherein the first logic derives each of the first and second indices by applying a hashing function to information derived from the resource request.

13. (Withdrawn) The system of claim 10 wherein the first logic derives the first index by applying a hashing function to a hashing key derived from a session or cookie identifier derived from a packet spawning the resource request.

14. (Withdrawn) The system of claim 10 where the second logic derives the second index by applying a hashing function to a hashing key derived from a client IP address derived from a packet spawning the resource request.

15. (Withdrawn) A system for making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the system comprising:

first means for deriving a first index from information relating to the resource request if such information is available, and using the first index to make an entry in the data structure representing the allocation if the first index is available; and

second means for deriving a second index from information relating to the resource request, and using the second index to make an entry in the data structure representing the

allocation.

16. (Withdrawn) A system for accessing a data structure in order to allocate a resource to a resource request based on application of a persistence policy, entries in the data structure corresponding to allocated resources, the system comprising:

first logic for deriving a first index from information relating to a resource request if such information is available, using the first index to access the data structure and determine if an entry corresponding to the first index is available, and, if such an entry is available, allocating the resource corresponding to the entry to the resource request; and

second logic for deriving, if the first index or an entry corresponding to the first index is unavailable, a second index from information relating to the resource request, and using the second index to access the data structure and determine if an entry corresponding to the second index is available, and, if such an entry is available, allocating the resource corresponding to the entry to the resource request.

17. (Withdrawn) The system of claim 16 further comprising third logic for allocating, if an entry corresponding to the second index is unavailable, a resource to the request based on application of a load balancing policy or other persistence policy.

18. (Withdrawn) The system of claim 17 further comprising fourth means for using the first index to make an entry in the data structure corresponding to the allocation of claim 17 if such first index is available.

19. (Withdrawn) The system of claim 18 further comprising fifth means for using the second index to make an entry in the data structure corresponding to the allocation of claim 17.

20. (Currently Amended) A method of allocating a resource in a network comprising: determining if any of a plurality of persistence policies, comprising at least one first persistence policy and at least one second persistence policy, in the network is applicable to a service request in the network, and, if so, allocating the resource to the request based on application of the persistence policy in the network determined to be applicable; and

when a first type of service request is received, the determining step comprises determining if the at least one first persistence policy is applicable;

when a second type of service request is received, the determining step comprises determining if the at least one second persistence policy is applicable;

when the first type of service request is received, but it is determined that the at least first persistence policy is inapplicable, the determining step comprises determining if the at least one second persistence policy is applicable; and

allocating the resource to the request based on application of a load balancing policy only if none of the persistence policies in the plurality of persistence policies in the network is determined to be inapplicable in the foregoing determining step;

~~wherein the persistence policy, if applicable, overrides the load balancing policy by controlling the allocation.~~

21. (Currently Amended) A method of allocating a resource, in a network, to a resource request based on application of a persistence policy, the request having an originator, comprising:

determining whether an allocation exists or recently expired for the originator of the resource request, based on application of any of a plurality of persistence policies, comprising at least one first persistence policy and at least one second persistence policy, to the request, and, if so, identifying the resource which is the subject of the existing or recently expired allocation; and

when a first type of service request is received, the determining step comprises determining whether an allocation exists or recently expired for the originator of the resource request by applying the at least first persistence policy, or the least one second persistence policy when the at least one first persistence policy is determined to be inapplicable;

when a second type of resource request is received, the determining step comprises determining whether an allocation exists or recently expired for the originator of the request by applying the at least one second persistence policy; and

allocating the resource in the network, once identified, to the resource request.

22. (Original) The method of claim 21 wherein the resource request is spawned by a packet.

23. (Withdrawn) A method of maintaining a data structure useful for allocating a resource to a resource request based on application of a persistence policy comprising:

- making an entry in the data structure representing an allocation, and time-stamping the entry with a time-stamp indicating the time when or about when the allocation is terminated; and
- scanning the data structure, and deleting entries for which a time-out condition is determined to exist.

24. (Withdrawn) The method of claim 23 further comprising determining that a time-out condition exists if the time-stamp value equals or exceeds a current time by a predetermined amount.

25. (Withdrawn) The method of claim 23 further comprising repetitively scanning the data structure.

26. (Withdrawn) The method of claim 23 further comprising periodically scanning the data structure.

27. (Withdrawn) The method of claim 24 wherein the predetermined amount is programmable.

28. (Withdrawn) A method of making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the method comprising:

- deriving a first index from information relating to the resource request if such information is available;

- using the first index to make an entry in the data structure representing the allocation if the first index is available;

- deriving a second index from information relating to the resource request; and

- using the second index to make an entry in the data structure representing the allocation.

29. (Withdrawn) The method of claim 28 wherein the data structure is a history table.

30. (Withdrawn) The method of claim 28 further comprising deriving each of the first and second indices by applying a hashing function to information derived from a packet spawning the resource request.

31. (Withdrawn) The method of claim 28 further comprising deriving the first index by applying a hashing function to a hashing key derived from a session or cookie identifier derived in turn from a packet spawning the resource request.

32. (Withdrawn) The method of claim 28 further comprising deriving the second index by applying a hashing function to a hashing key derived from a client IP address derived in turn from a packet spawning the resource request.

33. (Withdrawn) A method of making an entry in a data structure representing an allocation, the data structure being useful for allocating a resource to a resource request based on application of a persistence policy, the method comprising:

a step for deriving a first index from information relating to the resource request if such information is available;

a step for using the first index to make an entry in the data structure representing the allocation if the first index is available;

a step for deriving a second index from information relating to the resource request; and

a step for using the second index to make an entry in the data structure representing the allocation.

34. (Withdrawn) A method of accessing a data structure in order to allocate a resource to a resource request based on application of a persistence policy, entries in the data structure corresponding to allocated resources, the method comprising:

deriving a first index from information relating to a resource request if such information is available;

using the first index to access the data structure and determine if an entry corresponding to the first index is available;

if such an entry is available, allocating the resource corresponding to the entry to the resource request;

deriving, if the first index or an entry corresponding to the first index is unavailable, a second index from information relating to the resource request;

using the second index to access the data structure and determine if an entry

corresponding to the second index is available; and

if such an entry is available, allocating the resource corresponding to the entry to the resource request.

35. (Withdrawn) The method of claim 34 further comprising allocating, if an entry corresponding to the second index is unavailable, a resource to the request based on application of a load balancing policy or other persistence policy.

36. (Withdrawn) The method of claim 35 further comprising using the first index to make an entry in the data structure corresponding to the allocation of claim 35 if such first index is available.

37. (Withdrawn) The method of claim 36 further comprising using the second index to make an entry in the data structure corresponding to the allocation of claim 35.

38. (Withdrawn) A method of accessing a data structure in order to allocate a resource to a resource request based on application of a persistence policy, entries in the data structure corresponding to allocated resources, the method comprising:

a step for deriving a first index from information relating to a resource request if such information is available;

a step for using the first index to access the data structure and determine if an entry corresponding to the first index is available;

a step for allocating, if such an entry is available, the resource corresponding to the entry to the resource request;

a step for deriving, if the first index or an entry corresponding to the first index is unavailable, a second index from information relating to the resource request;

a step for using the second index to access the data structure and determine if an entry corresponding to the second index is available; and

a step for allocating, if such an entry is available, the resource corresponding to the entry to the resource request.

39. (New) The system of claim 1 wherein the at least one first persistence policy comprises at least one cookie-based persistence policy, and the at least one second persistence

policy comprises at least one client-based persistence policy.

40. (New) The system of claim 1 wherein the at least one first persistence policy comprises at least one session-based persistence policy, and the at least one second persistence policy comprises at least one client-based persistence policy.

41. (New) A system for allocating a resource in a network comprising:
first logic in the network for determining if any of a plurality of persistence policies, comprising at least one cookie-based persistence policy, at least one session-based persistence policy, and at least one client-based persistence policy, is applicable to a service request, and, if so, allocating the resource to the request based on application of the persistence policy determined to be applicable;

the first logic configured so that:

when a first type of service request is received, the first logic determines if either the at least one cookie-based or at least one session-based persistence policy is applicable;

when a second type of service request is received, the first logic determines if the at least one client-based persistence policy is applicable;

when the first type of service request is received but it is determined that both the at least one cookie-based persistence policy and the at least one session-based persistence policy are inapplicable, the first logic determines if the at least one client-based persistence policy is applicable; and

second logic in the network for allocating the resource to the request based on application of a load balancing policy only if none of the plurality of persistence policies is determined to be applicable as determined by the first logic.

42. (New) The system of claim 1 wherein the first type of resource request is content-aware, and the second type of resource request is non-content-aware.

43. (New) The system of claim 42 where the at least one first persistence policy is content-aware, and the at least one second persistence policy is non-content-aware.